



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

---

## Technical Documentation: Smart Greenhouse

### Project Details:

- **Title:** Discover the Green Life with Robots.
- **Project ID:** 2023-1-HU01-KA210-VET-000156243.
- **Program:** Erasmus+.
- **Created:** 2024.
- **Partner Institutions:** Istanbul Gülizar Zeki Obdan Anadolu Lisesi (Turkey) and Székesfehérvári SZC Széchenyi István Műszaki Technikum (Hungary).

**Funding Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

## 1. Introduction

Within the framework of the Erasmus+ project bearing the code "2023-1-HU01-KA210-VET-000156243," students from the partner institutions developed and built a small-sized, automated smart greenhouse.

The goal of the project was to create an **autonomous climate control system** capable of maintaining the internal environmental parameters (temperature, humidity, light intensity, soil moisture) at the set optimal values.

### 1.1 Hardware Basics

The basis of the greenhouse was a plexiglass baby incubator, which was used 20 years ago. This incubator provided an ideal, enclosed, and transparent environment for two potted ornamental plants. The control unit, which contains the **Arduino Mega microcontroller**, was placed in a box designed in Tinkercad.

## 2. System Components and Functions

The smart greenhouse system can be divided into two main parts: **sensors (input)** and **actuator units (output/actors)**. These units are all controlled by the Arduino Mega. The



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

adjustment of the set values was realized using a rotary encoder, with feedback displayed on a two-row LCD.

## 2.1 Sensors (Input Units)

The following sensors were connected to the Arduino Mega for data collection:

Sensor	Function	Location	Connection
Temperature/Humidity sensor (DHT)	Measurement of the two basic internal climate parameters.	Incubator interior.	Arduino Mega Analog/Digital input.
Light sensor	Measurement of the internal illumination intensity.	Incubator interior.	Arduino Mega Analog input.
2 Soil Moisture sensors	Measurement of the soil moisture of the two ornamental plants.	In both flower pots.	Arduino Mega Analog input.

## 2.2 Actuator Units (Actors)

The Arduino controls the climate-providing systems based on the sensed data.

Actor	Function	Control	Operation Description
2 5V Pumps	Providing watering.	Arduino controlled relay.	Based on the signal from the soil moisture sensor.
2 Computer Power Supply Fans	Ventilation.	Arduino controlled relay.	For the purpose of humidity and/or temperature regulation.
Full Spectrum LED strip	Lighting.	Arduino controlled relay.	Glued to the inner top surface of the greenhouse, controlled by a light sensor.
Mini Hair Dryer	Heating.	Arduino controlled relay.	For the purpose of temperature regulation.

## 3. Circuit Plan and Connections



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

The actuator units (pumps, fans, LED strip, hair dryer) are controlled via **relay modules** because they typically require a higher current than the Arduino digital outputs can directly provide.

Component	Arduino Mega Pin	Type	Description
Two-row LCD	I2C (SDA, SCL) or Digital D22–D29.	Output.	Data display.
Rotary Encoder (CLK/DT)	Digital D2, D3.	Input (Interrupt- compatible).	Menu management, value setting.
Light Sensor	Analog A0.	Input.	Measuring light intensity.
Temperature/Humidity (DHT)	Digital D4.	Input.	Measuring climate data.
Moisture 1	Analog A1.	Input.	Control of watering 1.
Moisture 2	Analog A2.	Input.	Control of watering 2.
Pump 1 (Relay)	Digital D5.	Output.	Control of watering 1.
Pump 2 (Relay)	Digital D6.	Output.	Control of watering 2.
Fan 1 & 2 (Relay)	Digital D7.	Output.	Ventilation (connected in parallel).
LED strip (Relay)	Digital D8.	Output.	Control of lighting.
Mini Hair Dryer (Relay)	Digital D9.	Output.	Control of heating.

#### 4. Operating Principle

The system operates cyclically, continuously reading sensor data and comparing it to the target values set by the rotary encoder. The Arduino immediately reacts to changes in the internal environment based on the input signals.



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

1. **Watering (Soil Moisture):** If the value on analog input A1 or A2 falls below the set threshold (indicating dry soil), the appropriate 5V pump activates (D5 or D6 relay) until the moisture level is restored.
2. **Temperature Regulation:** If the measured temperature is too low, the mini hair dryer (D9 relay) providing heating activates. If the temperature is too high, the ventilation fans activate (D7 relay).
3. **Humidity Regulation:** If the air humidity is too high, the ventilation fans activate (D7 relay) to reduce the humidity.
4. **Lighting Regulation:** The full spectrum LED strip switches on and off based on the set light requirement and the light sensor measurement (D8 relay).

## 5. Program Code (Arduino Mega)

The following code implements the functions described and assumes the necessary external libraries (e.g., LiquidCrystal\_I2C, DHT, RotaryEncoder) are available.

```
#include <LiquidCrystal_I2C.h>

#include <DHT.h>

#include <RotaryEncoder.h>

// PIN DEFINITIONS

#define DHT_PIN 4

#define DHT_TYPE DHT22 // Assumed

#define POT_MOISTURE_1 A1

#define POT_MOISTURE_2 A2

#define RELAY_PUMP_1 5

#define RELAY_PUMP_2 6

#define RELAY_FAN 7

#define RELAY_LED 8
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
#define RELAY_HEATER 9
```

```
#define LIGHT_SENSOR A0
```

```
// ROTARY ENCODER PINS
```

```
#define ROTARY_CLK 2
```

```
#define ROTARY_DT 3
```

```
#define ROTARY_SW 10 // Assumed for menu selection
```

```
// Initialization
```

```
DHT dht(DHT_PIN, DHT_TYPE);
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); // Assumed I2C address
```

```
RotaryEncoder encoder(ROTARY_CLK, ROTARY_DT, RotaryEncoder::LatchMode::FOUR3);
```

```
// Target States (Adjustable with Rotary Encoder)
```

```
int targetTemp = 25;
```

```
int targetHumidity = 60;
```

```
int targetMoisture = 400; // Analog value (0-1023)
```

```
int targetLight = 500; // Analog value (0-1023)
```

```
// Current values
```

```
float currentTemp = 0.0;
```

```
float currentHumidity = 0.0;
```

```
int currentMoisture1 = 0;
```

```
int currentMoisture2 = 0;
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
int currentLight = 0;
```

```
void setup()
```

```
{
```

```
  // Serial communication (for debugging)
```

```
  Serial.begin(9600);
```

```
  // LCD and DHT initialization
```

```
  lcd.init(); lcd.backlight();
```

```
  dht.begin();
```

```
  // Actuators (Relays) set to output
```

```
  pinMode(RELAY_PUMP_1, OUTPUT);
```

```
  pinMode(RELAY_PUMP_2, OUTPUT); pinMode(RELAY_FAN, OUTPUT);
```

```
  pinMode(RELAY_LED, OUTPUT);
```

```
  pinMode(RELAY_HEATER, OUTPUT);
```

```
  pinMode(ROTARY_SW, INPUT_PULLUP);
```

```
  // Setting default state (all off)
```

```
  digitalWrite(RELAY_PUMP_1, HIGH); // Assuming NO (Normally Open) relay
```

```
  digitalWrite(RELAY_PUMP_2, HIGH);
```

```
  digitalWrite(RELAY_FAN, HIGH);
```

```
  digitalWrite(RELAY_LED, HIGH);
```

```
  digitalWrite(RELAY_HEATER, HIGH);
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
    lcd.setCursor(0, 0); lcd.print("Smart Greenhouse");

    lcd.setCursor(0, 1); lcd.print("Ready...");

    delay(2000);
}

void readSensors() {
    currentTemp = dht.readTemperature();
    currentHumidity = dht.readHumidity();
    currentMoisture1 = analogRead(POT_MOISTURE_1);
    currentMoisture2 = analogRead(POT_MOISTURE_2);
    currentLight = analogRead(LIGHT_SENSOR);

    if (isnan(currentTemp) || isnan(currentHumidity))
    {
        Serial.println("DHT error!");
    }
}

void handleClimateControl() {
    // 1. Temperature control (Mini Hair Dryer and Fan)
    if (currentTemp < targetTemp - 2) {
        digitalWrite(RELAY_HEATER, LOW); // Heating ON
    } else if (currentTemp > targetTemp + 1 || currentHumidity > targetHumidity + 5) {
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
        digitalWrite(RELAY_FAN, LOW); // Cooling/Ventilation ON

        digitalWrite(RELAY_HEATER, HIGH); // Heating OFF

    } else {

        digitalWrite(RELAY_HEATER, HIGH);

        digitalWrite(RELAY_FAN, HIGH);

    }

// 2. Lighting control (LED strip)
if (currentLight < targetLight) {

    digitalWrite(RELAY_LED, LOW); // Lighting ON

} else {

    digitalWrite(RELAY_LED, HIGH); // Lighting OFF

}

}

void handleWatering() {

    // Soil Moisture 1 (Watering 1)

    if (currentMoisture1 > targetMoisture) { // Higher value = drier soil

        digitalWrite(RELAY_PUMP_1, LOW); // Pump ON

        delay(3000); // Watering time (needs calibration)

        digitalWrite(RELAY_PUMP_1, HIGH); // Pump OFF

    }

    // Soil Moisture 2 (Watering 2)
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
if (currentMoisture2 > targetMoisture) {  
    digitalWrite(RELAY_PUMP_2, LOW);  
    delay(3000);  
    digitalWrite(RELAY_PUMP_2, HIGH);  
}  
  
}  
  
void displayStatus() {  
    lcd.clear(); lcd.setCursor(0, 0);  
    lcd.print("T:");  
    lcd.print(currentTemp, 1);  
    lcd.print("C H:");  
    lcd.print(currentHumidity, 0);  
    lcd.print("%");  
  
    lcd.setCursor(0, 1);  
    lcd.print("N1:");  
    lcd.print(1023 - currentMoisture1); // Inverse logic, higher value = wetter soil  
    lcd.print(" N2:");  
    lcd.print(1023 - currentMoisture2);  
}  
  
void loop() {  
    readSensors();
```



**Project ID:** 2023-1-HU01-KA210-VET-000156243

**Disclaimer:** Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Tempus Public Foundation. Neither the European Union nor the granting authority can be held responsible for them.

```
handleClimateControl();
```

```
handleWatering();
```

```
displayStatus();
```

```
// Menu management (rotary encoder) – This part is currently undeveloped for code  
simplification,
```

```
// but this is where the target values (targetTemp, targetHumidity, etc.)
```

```
// should be implemented for adjustment using the rotary encoder and displayed on the  
LCD.
```

```
delay(5000); // 5 second cycle time
```

```
}
```